[Lecture 1] Basic Principles

Why python?

- Popular
- Big standart library
- Huge set of third-party libraries
- Rich set of powerful language-level data structures
- Easy to learn, easy to program

Characteristics:

- High-level language (есть переменные, типы, выражения,)
- Object-oriented language
- Functional language
- Modular language (собирается из модулей)
- Dynamically typed (динамическая типизация)
- Interpreted language (считывается построчно)

Static typing:

— Сложнее писать(нужно задавать типы)

+ More safe program

+ More readable program

Dynamic typing:

+ Легче писать

+ More flexible variables usage

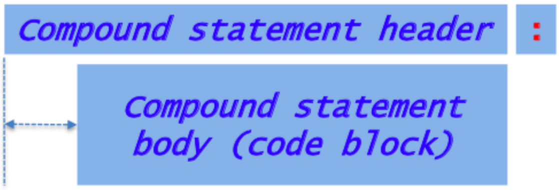
— Сложна для понимания

— Unsafe and inefficient

Indentation - отступ (пробелы)

Compound statement - составной оператор

Common syntax(не обязательный) —→


Compound statement header
:
Compound statement body (code block)

Execution model:

Interpretation(line-by-line) - transforms code to IR (byte code) and executes it

Input(program) can be specified as complete module (real programming) or can be typed interactively (shell experimenting)

1. Everything is an object - variables, functions, classes (!), modules (!!).

2. Each "variable" is a reference. (variable →(ref) object)

Assignment manipulates references

a = 1; b=a ;(b = 1)

a = a+1 (a = 2; b = 1)

Python functions: OBJECTS TOO!!!

Usage: if you need a portion of code to be executed several times, for various input values

Actions:

1. Call / Invoke / Launch
2. Assign to a variable
3. Pass as argument to another function
4. Return as result of another function

f - reference to an object of functional type

f(7) - call operator applied to function